Self-Repairing Embryonic Memory Arrays

Lucian Prodan

Mihai Udrescu

Mircea Vladutiu

"Politehnica" University of Timisoara
ROMANIA

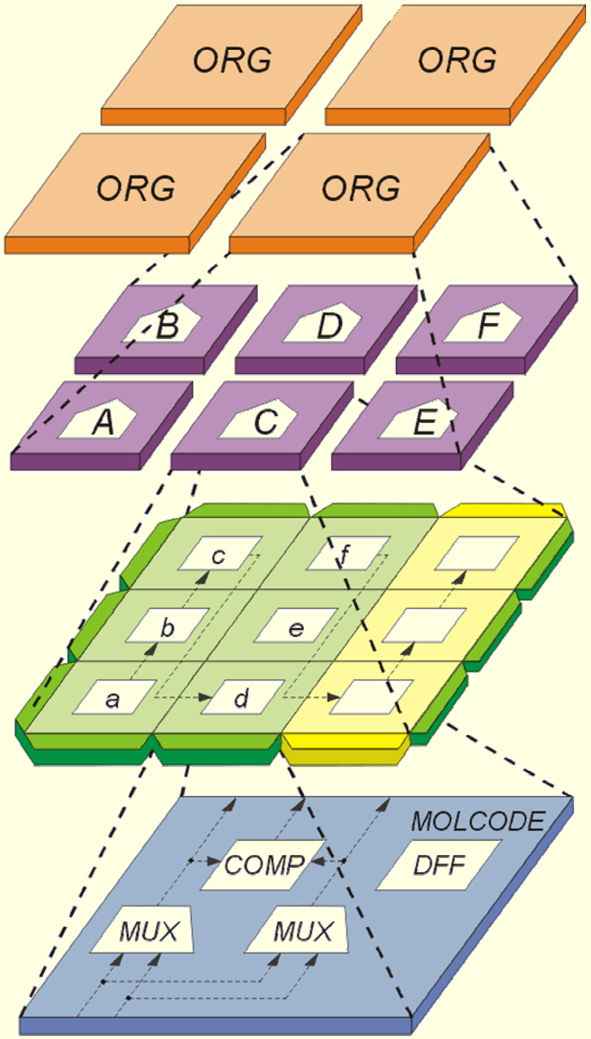
June 24-26, 2004

NASA/DoD IEEE Conference
on Evolvable Hardware

Seattle

What is Embryonics?

- Bio-inspired computing system
- Aimed at transferring biological robustness into digital electronics
- Four-level system architecture hierarchy
- Hierarchical self-repairing



Population level

Organism level

Cellular level

Molecular level

The Genetic Program

- Cells delimited by **polymerase genome** **(the cellular membrane or** **space divider)**
- Molecules configured by **ribosomic genome**
- Two operating modes possible for a molecule
  - Logic mode: a functional unit based on two multiplexers and a flip-flop, together with signal routing mechanism to and from neighbors
  - Memory mode: program called **operative genome**

The Memory Mode

Genetic program stored by each molecule in pieces of either 8 bits or 16 bits

Memory structures are made of molecules, are delimited by a membrane mechanism, but are not cells ⇨ **macro-molecules**

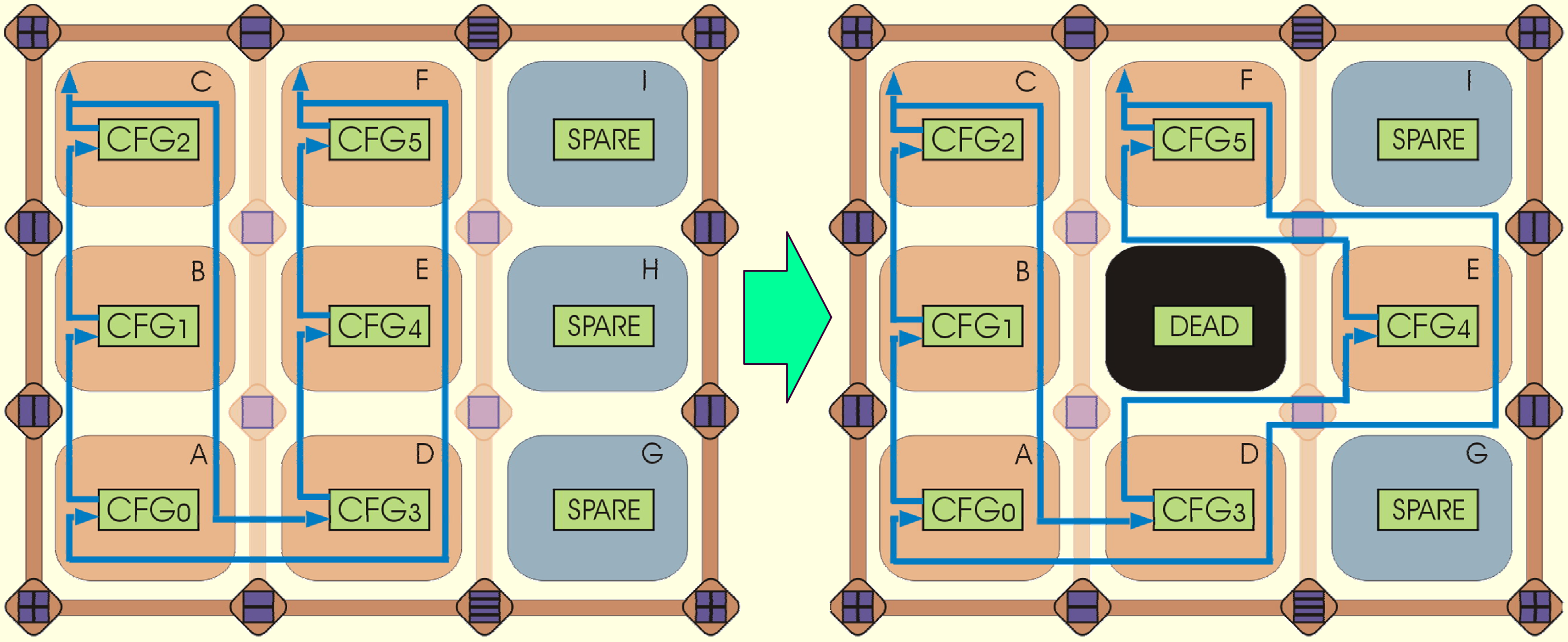
Memory molecules from within the same macro-molecule are all chained together

Data is shifted continuously ⇨ **cyclic-type memory**

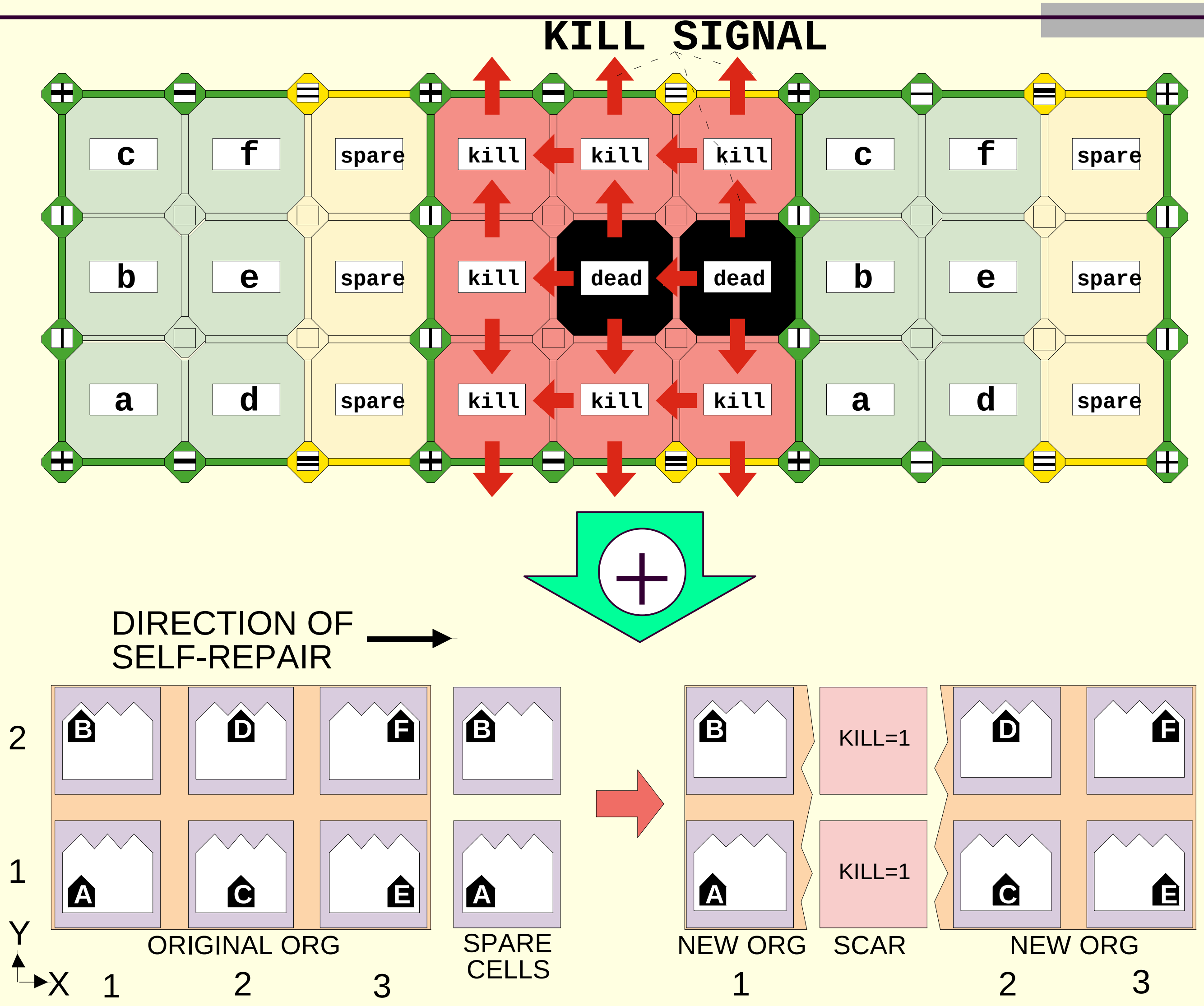
Molecular Self-Repair (Logic Mode)

A faulty molecule is replaced with a spare one, by transferring its functionality

The faulty molecule is then disabled, i.e. “dies”



Hierarchical Self-Repair



KILL SIGNAL

c	f	spare	kill	kill	kill	c	f	spare
b	e	spare	kill	dead	dead	b	e	spare
a	d	spare	kill	kill	kill	a	d	spare

DIRECTION OF SELF-REPAIR

Molecular Self-Repair (Memory Mode)

Functionality transfer not possible in memory mode

Transferring genetic data from a faulty molecule to a spare one also transfers the fault(s), thus wasting valuable spare resources

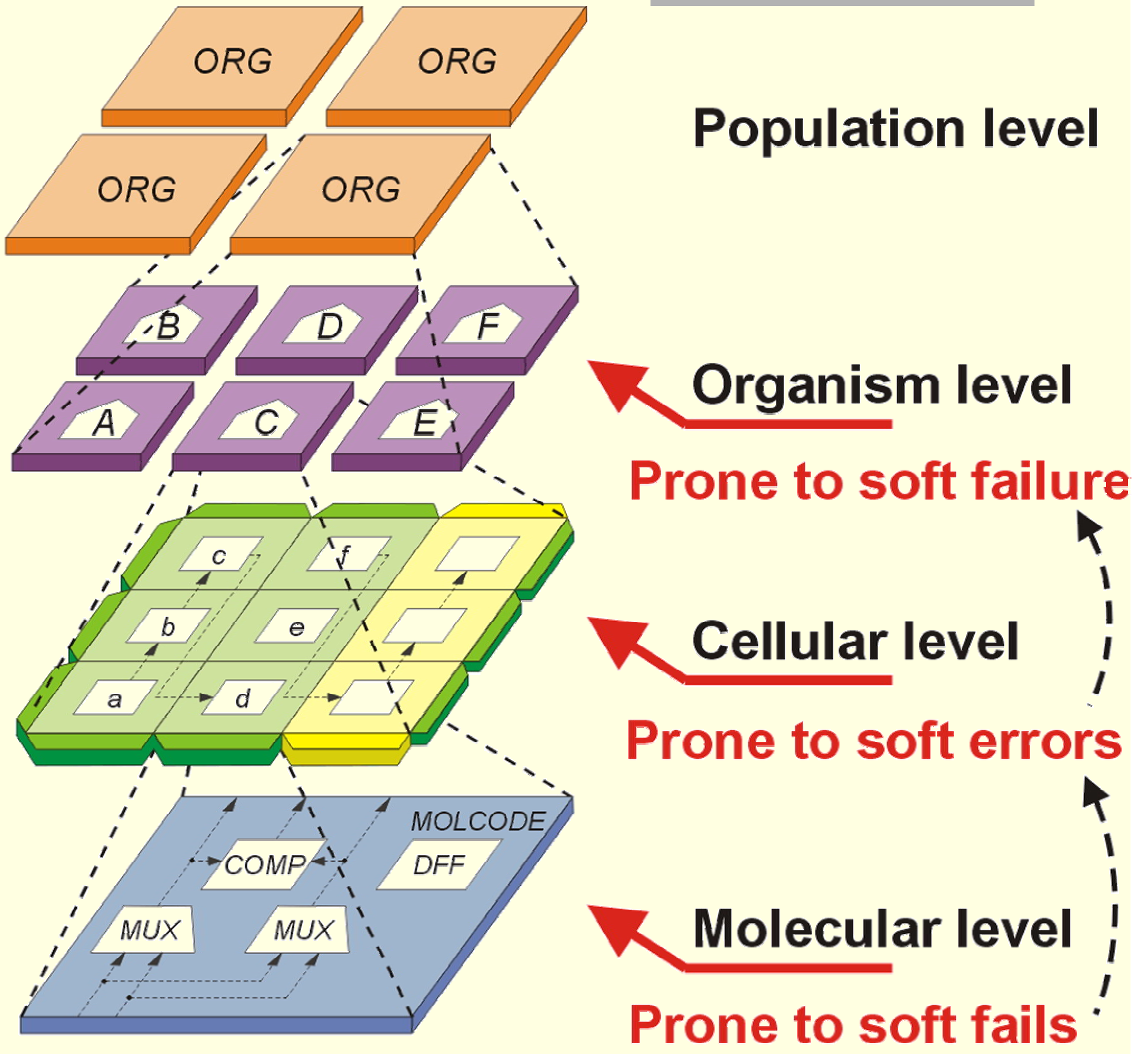
↓

Existent self-repair mechanism therefore not able to ensure protection for macro-molecules

Memory Vulnerability

Memory affected by soft fails

Soft fails: transient errors induced by energized atomic particles that hit a semiconductor device



Population level

Organism level

Prone to soft failure

Cellular level

Prone to soft errors

Molecular level

Prone to soft fails

Origins of Soft Fails

- Human expansion into space bound to **aggressive radiation** exposure
- Experiments attempting to measure particle flux since 1980 (IBM)
- Three categories of radiation
  - Primary cosmic rays: eventually may hit our planet; mostly protons (92%) and aŁparticles (6%)
  - Cascade particles, born form collisions when primary cosmic rays enter the earth’s atmosphere
  - Terrestrial cosmic rays: energetic particles reaching the surface; mostly cascade-generated; only 1% due to primary cosmic rays

Soft Errors

- *„by far the most common type of chip failure is a soft error of a single cell on a chip“*
- Main cause for memory protection techniques: mitigation measures (physical level), parity codes, Error Checking and Correcting or ECC (data level)
- Two issues concerning protective techniques for memory devices:
  - Error detection (low HW overhead)
  - Error correction (greater HW overhead but superior effectiveness)

Soft Error Rate

Chip type	Observed SER	Typical application
4Kb bipolar	1.340	Cache memory
288 Kb DRAM	126.000	Main memory
1Mb DRAM	3.000	Main memory
144Kb CMOS	210	Secondary cache
9Kb bipolar	998	I/O channels

Soft Error Rates for a variety of IBM memory chips show the effect of radiations over semiconductor devices

Embryonics

- Robustness transfer from biology in Embryonics project hampered by memory vulnerability
- Genetic program protected in biological entities; DNA capable of detecting and correcting a variety of faults
- If Embryonics is to claim bio-inspired robustness, memory protection for most frequent upsetting scenario is a must

Reliability Analysis

Following scenarios possible:

Fault tolerance at the molecular level; Advantage: isolate the faulty molecule, use the self-repair mechanism already in place; Disadvantage: HW overhead

Fault tolerance at the macro-molecular level; Advantage: ECC coding, lower HW overhead; Disadvantage: no use for the existent self-repair mechanisms

Memory Reliability w/o FT (1)

Macro-molecular dimensions: M lines, N columns, s spare columns

Each molecule stores F bits of genome data

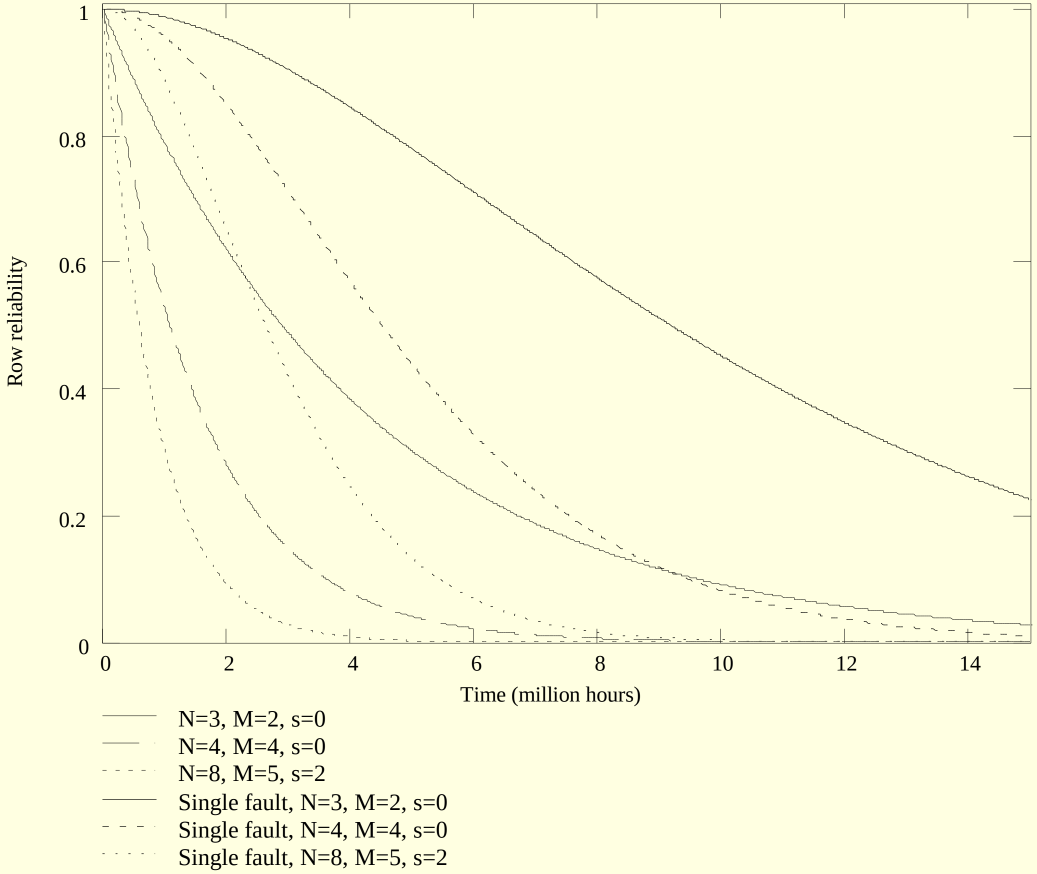
Failure rate for a storage flip-flop ?Õ

ΔT_{med} mean period between two consequent upset events inside the macro-molecular area

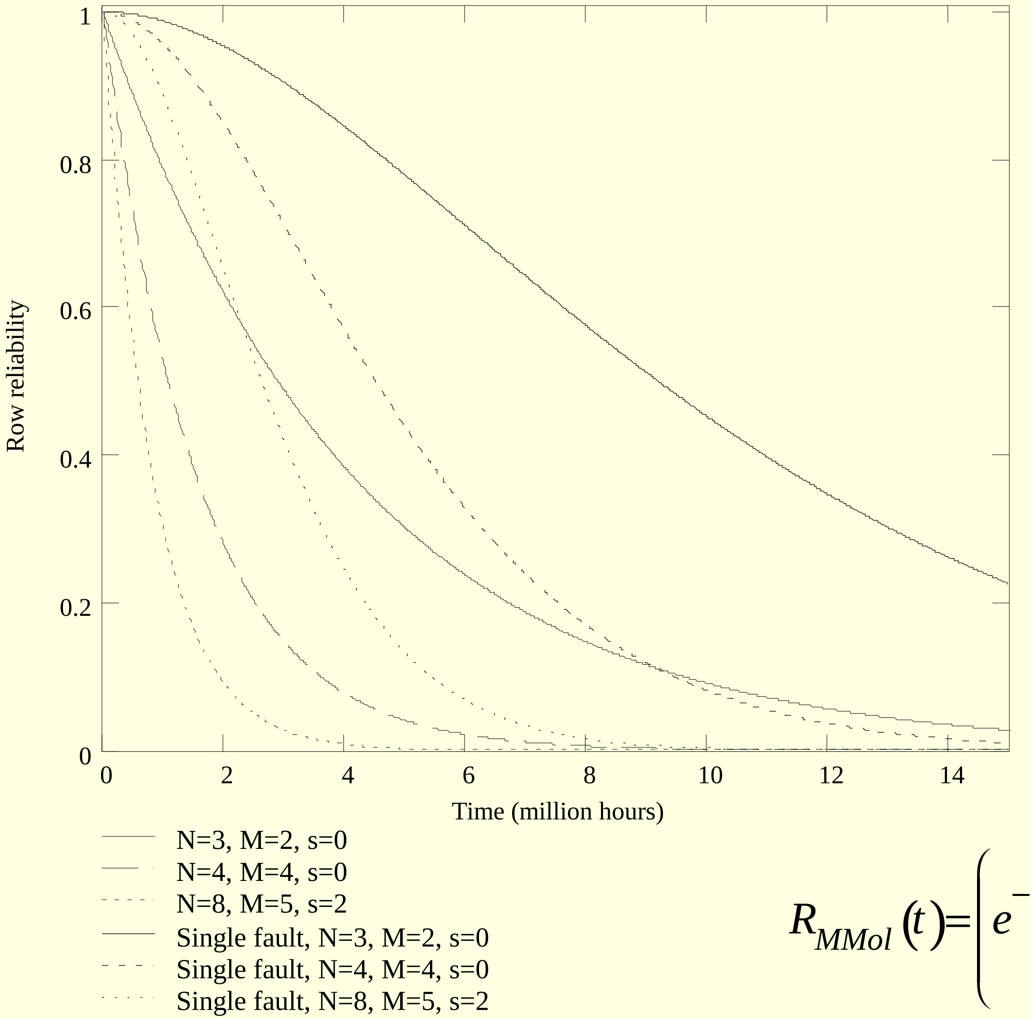
R(t)=Prob{unrecoverable error has not yet occurred}

Memory Reliability w/o FT (2)

$$R_{MMol}(t) = e^{-\lambda FM(N-s)t} \qquad MTTF_{MMol} = \min\left(\int_{0}^{\infty} R_{Row}^{M}(t)\,dt,\ \Delta\mathrm{T}_{med}\right)$$



FT at the Molecular Level



$$MTTF_{MMol} = \min\left(\int_0^\infty R_{Row}^M(t)\,dt,\ 2\Delta T^*_{med} \right)$$

$$\Delta T^*_{med} = M\,\Delta T_{med}$$

$$R_{MMol}(t) = \left(e^{-\lambda F(N-s)} + F(N-s)\left(1-e^{-\lambda t}\right)e^{-\lambda t\left(F(N-s)-1\right)} \right)^M$$

The Failure Rate ?l

?áessentially an empirical parameter

Value determined by extensive measurements

Exposure to aggressive environments affects ?•values

From a **constant parameter** (at sea-level and during standard environment conditions), ?•becomes a **variable** (at high altitudes or in outer space, during non-standard conditions).

Fault Tolerant Memory Structures

Overall reliability increased by two fundamental techniques:

Fault prevention (aka fault intolerance) eliminates possible faults at the initial moment; already present in Embryonics

Fault tolerance allows valid computations through redundancy, even in the presence of faults; not present in Embryonics, subject of this paper

Fault Tolerance and Embryonics

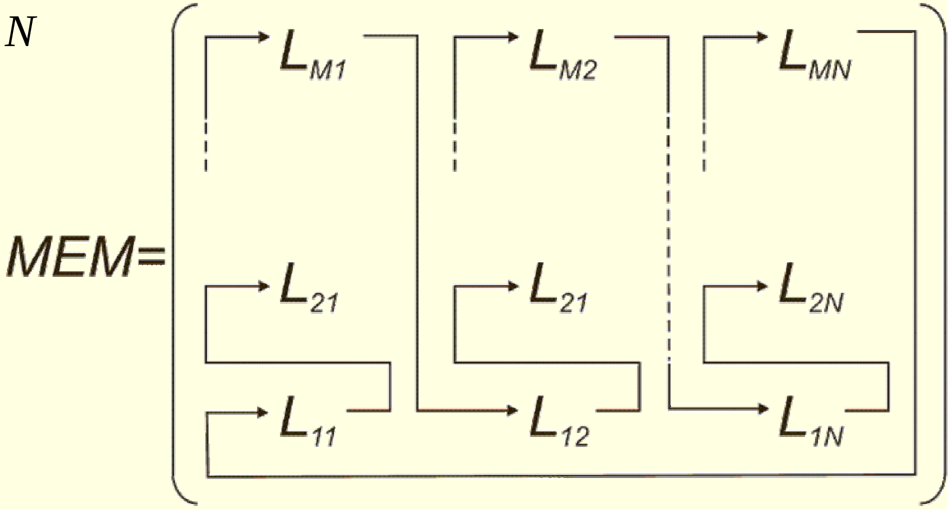
- Only the functional part of the molecule is currently fault tolerant
- The addition of memory molecules not covered:
  - no error detection inside a memory molecule
  - self-repairing mechanism overcome, preserving erroneous data; resource wasting while offering no data protection

↓

ECC implementation necessary

Memory Datapath

$$V\left(L_{i,j}\right)=\begin{cases}\left(L_{i\text{-}1,j}\,L_{i,j}\,L_{i+1,j}\right), & \text{if } 1<i<M,\ 1<j<N\\ \left(L_{M,j-1}\,L_{1,j}\,L_{2,j}\right), & \text{if } i=1,\ 1<j\\ \left(L_{M,N}\,L_{1,1}\,L_{2,1}\right), & \text{if } i=j=1\end{cases}$$



$$\tilde{L}_{i,j}=L_{i,j}\left(t+1\right)=V\left(L_{i,j}\left(t\right)\right)\times M^{*}$$

$$M^{*}=\begin{pmatrix}0_{\left(F-1\right)\times F}\\ I_{F}\\ 0_{\left(F+1\right)\times F}\end{pmatrix}$$

$$E\left(L_{ij}\right)\left(k\right)=\begin{cases}1, & \text{if bit k is erroneous}\\ 0, & \text{otherwise}\end{cases}$$

$$V\left(L_{i,j}\right)=\left(L_{6,3}\quad L_{1,4}\quad L_{2,4}\right)$$

$$\tilde{L}_{i,j}=\left(V\left(L_{i,j}\right)\oplus E\left(L_{i,j}\right)\right)\times M^{*}$$

Example

Genome data words 4-bit-wide → (4,7) code

Final structure for a FT macro-molecule:

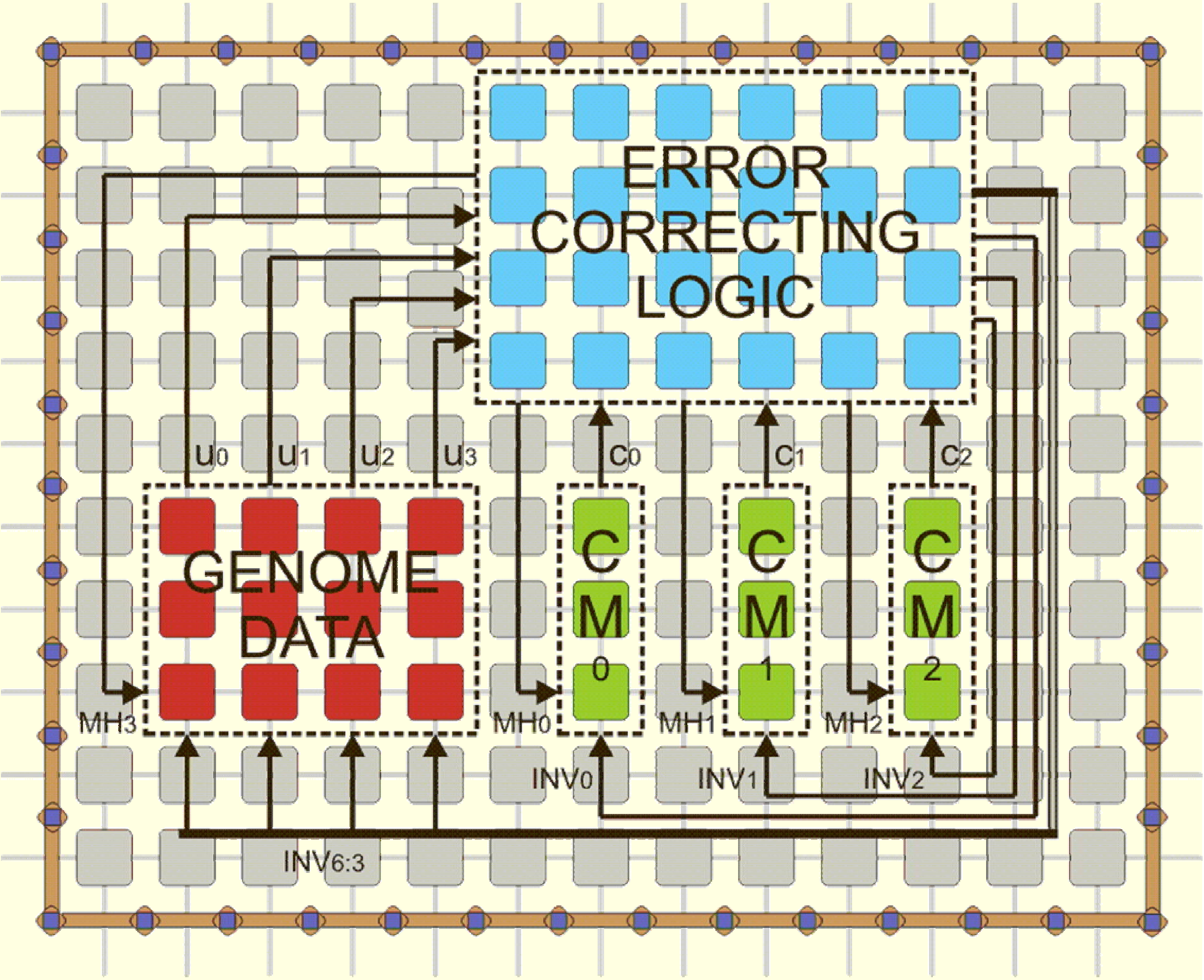
- Data macro-molecule
- 3 macro-molecules for check data
- Additional error checking and correcting logic

Additional signals required:

- Memory Hold – enables data shifting for a macro-molecule
- INVert – enables data correction

Implementation

- Protection for single errors (most frequent)
- Based on Hamming-class codes
- Multiple error detection possible



Control Signals

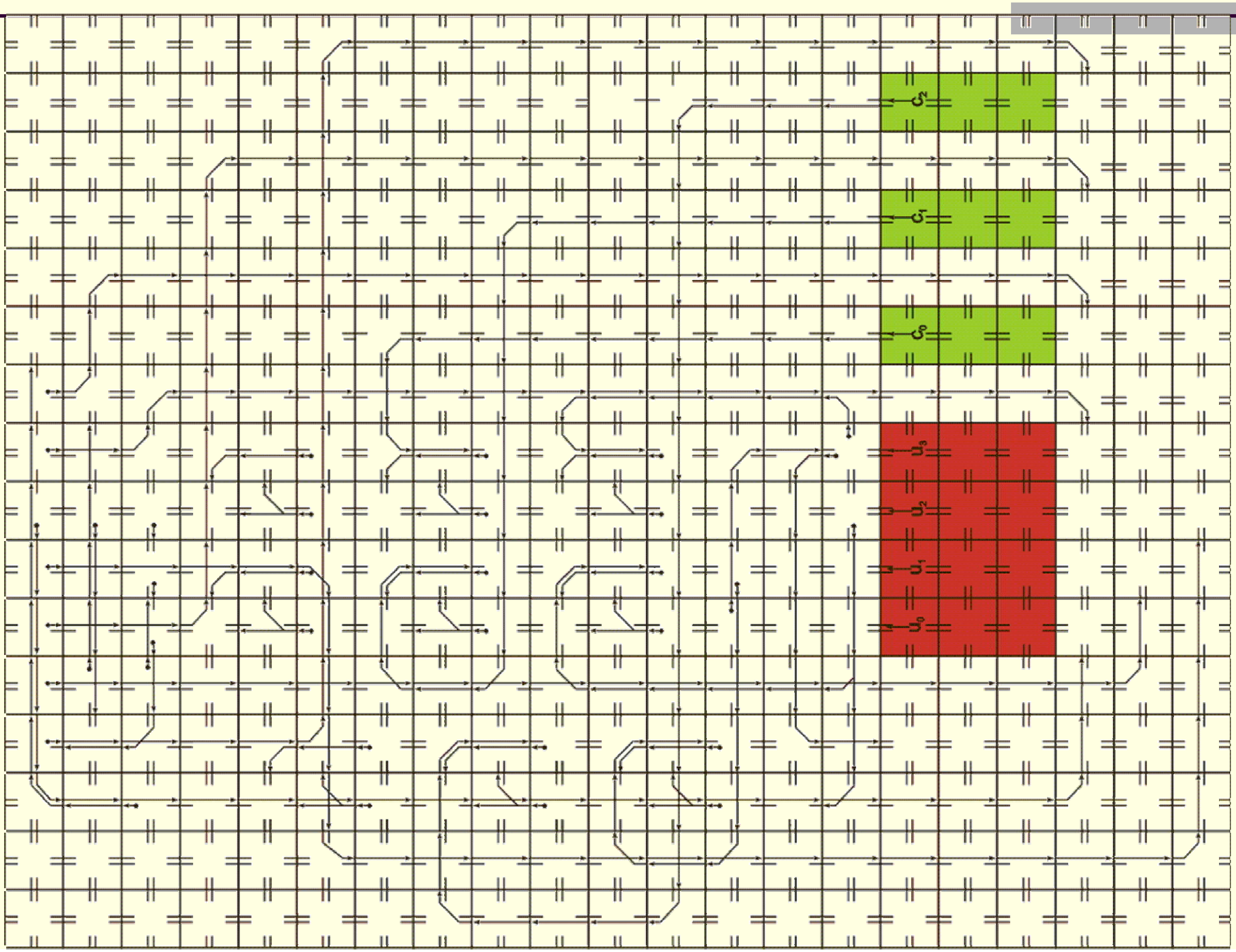
MH_i	$INV_{0\ 1\ \ldots\ n-1\ n}$	Operation
0	11 … 11	Memory shift enabled
0	01 … 11	Memory shift with column 0 inverted
0	10 … 11	Memory shift with column 1 inverted
M	M	M
0	11 … 01	Memory shift with column n-1 inverted
0	11 … 10	Memory shift with column n inverted
1	xx … xx	Memory shift disabled

Final Design: Resource Levels

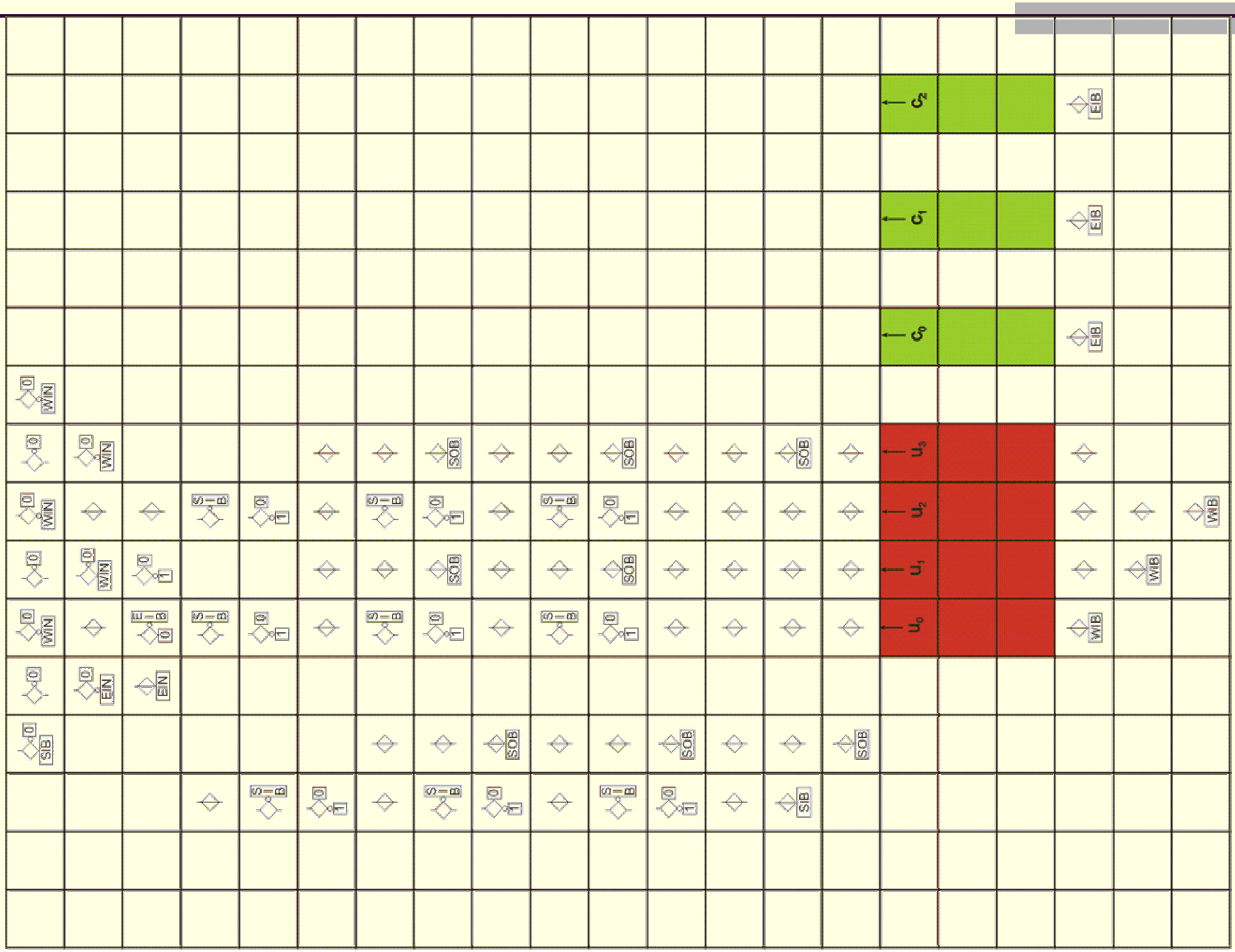
Two levels of configuration:

- Bus level – contains routing information for all buses
- Logic level – configures the Functional Unit and CREG for each molecule

The Bus Level



The Logic Level

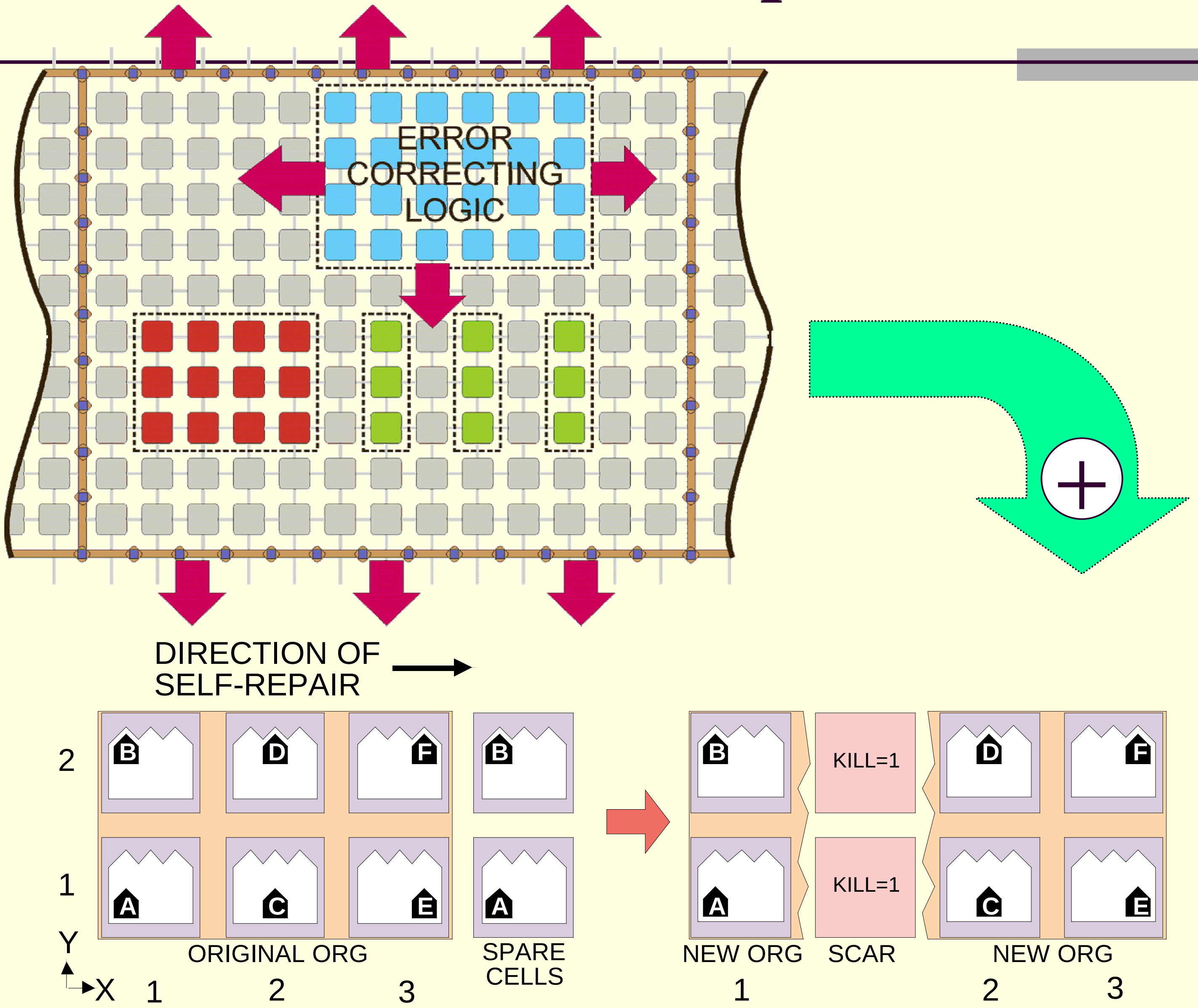


Self-Repairing Macro-Molecules

At the ***molecular level***, single faults are detected and corrected by the Error Correcting Logic

If an occurring fault has been detected but cannot be corrected, the Error Correcting Logic triggers the KILL signal, which activates the self-repair at the ***cellular level***

Hierarchical Self-Repair



Conclusions and Future Work

- ü Two-level self-repair now covering the memory structures
- ü Additional logic proportionally smaller when larger macro-molecules used

- ? Model for automatic fault tolerance assessment
- ? Design techniques with “Embryonics FPGA”
- ? …